Address Correlation: Exceeding the Limits of Locality

Resit Sendag
University of Rhode Island

Peng-fei Chuang, David J. Lilja
University of Minnesota

Introduction
Where did the idea come from?

- Store value locality
  - Redundant (silent) stores
  - Same data written to many addresses
- Frequent value locality
  - A few values appear very frequently in memory locations
- Address Correlation
  - Link the addresses that reference the same data

Why Address Correlation?

- Service a load by another (correlated) address on
  - A cache miss
  - A partial hit
- How much potential does it have?
  - Potential candidates for address correlation
  - Profiling results
Potential of an Address Correlation System

The percentage of all load misses and partial hits whose values are found in other addresses. The L1 data cache is 32KB and 4-way associative.

Potential Upper-bound Speedup

The percentage speedup by supplying data from a potential address residing in L1 cache.

How many addresses can supply data on a miss

The number of different addresses in cache, in which the data miss is found.

Where is the data found?

The distance of the address in which (missed) data is found in cache (measured by sets)
Investigating the Program Behavior

**Correlation**

- between fields of structures
- between references to instances
- of frequent values

Correlation between fields of structures

- In many instances of a structure, there often are fields that contain the same values.
- Example: A database of students in a high school
  - Student (state, city, etc.)
- Examples can be found in SPEC 2000 benchmarks

Correlation between references to instances

(a) Two different molecules using some of the same component atoms.

(b) The user-defined type in `AMM` for storing the angle information.

The values of references to `angle1.atom1`, `angle1.atom3` and `angle2.atom1` to be the same.

Correlations of frequent values

- Another major source of address correlation
- Intuitively, the more copies of a value that exist in the cache, the more often useful correlations can be created.
- For example, “zero” is extensively used
  - for variable initialization,
  - for constants such as NULL or FALSE,
  - to fill sparse matrices, and
  - as the starting value of enumeration types.
- Important fraction of the correlated addresses come from frequent values such as zero.
**Upper-bound Potential**

The normalized cache miss counts for varying L1 data cache sizes with and without address correlation (ac). The original (org) superscalar processor with an 8KB, 4-way associative L1 data cache is used as the base for these comparisons.

**Limiting the number of correlations**

The percentage of L1 data cache misses eliminated at run-time by an ACS. The L1 data cache is 32KB with 4-way associativity. 62% of the misses can be eliminated on average and 68% of the potential addresses can be correlated at run-time.

---

**Conclusions**

- A new approach for exploiting value locality
- Introduced Address correlation
  - Link the addresses that reference the same data
  - Supply the data miss or a partial hit by a correlated address
- Sources of Address correlation
  - semantically equivalent information
  - duplicated references
  - frequent values

**Future Work**

- Develop a feasible implementation
  - Search only nearby locations in the cache
  - The amount of useful correlation is usually bounded
  - Profiling results show that usually 1-2 links are enough